language ("for example," "such as"). Applicant respectfully traverses this rejection. It appears that the Examiner may have misread these claims, since the claims recite a step of "associating," rather than "associated." The verb "associate" means simply "to join (things) together or connect (one thing) with another," as defined in *Webster's Third New International Dictionary*, and "associating" is the gerund form of this verb. Claim 7 recites that comparing the tableau transitions comprises <u>associating</u> model transitions with tableau transitions, while claim 12 recites the added step of <u>associating</u> model states with tableau states. The language of claims 7-9 and 12 is thus based on clear, standard English usage, and there is nothing exemplary about it. Therefore, Applicant believes that the rejection under 35 USC 112 should be withdrawn.

Claims 1-32 were rejected under 35 USC 103(a) over Cleaveland ("Tableau-Based Model Checking…") in view of Cleaveland et al. (U.S. Patent 6,385,765). Applicant respectfully traverses this rejection.

Claim 1 in the present patent application recites a method for verification, which may be used to evaluate how well a specification of a set of properties for use in model checking covers the model that it is supposed to check. In model checking, a verification engineer writes a set of properties (known as a *specification*) that a design is expected to fulfill. An *implementation model* is then tested to ascertain that the model satisfies all of the properties in the set (page 1, lines 14-28). Before the present invention, however, the verification engineer had no systematic way to be sure of whether the property set is complete (page 2, lines 14-18). If the property set is incomplete, a bug in the design may go undetected.

In the solution recited by claim 1, this problem is addressed by creating a tableau from the specification.

A "*tableau*" is defined in the present patent application as "a finite state machine that satisfies all of the properties in the specification" (page 4, lines 14-16). Thus, as recited in the claim, the tableau comprises "tableau states" with "tableau transitions" between the states. The tableau transitions are compared to the model transitions of the implementation model in order to determine whether there is any discrepancy between the two. "To the extent that there are no substantive differences, it is concluded that the set of properties fully specifies the model" (page 4, lines 20-22).

Cleaveland describes a procedure for model checking using a type of logic known as "mu-calculus." As noted in the abstract, Cleaveland uses a "tableau-based proof system" to prove that this type of logic is "sound and complete," i.e., that the mu-calculus itself is suitable for use in model checking. He gives an example of a "tableau," consisting of rules R1-R8, in Figure 3 on page 6. He then defines a tableau as follows (page 7, lines 9-10): "a tableau for σ is a maximal proof tree having σ as its root and constructed using R1-R8." This "tableau," in other words, is a hierarchy of rules and has nothing to do with a state machine. Unlike the tableau recited in claim 1, <u>Cleaveland's tableau has no tableau states and no tableau transitions</u>. The only states and transitions that Cleaveland does discuss (page 3, line 21 - page 4, line 2) are those of the model itself, not of the tableau.

Cleaveland et al. describe a method for specifying concurrent systems (abstract). One aspect of the method involves model checking, which "allows one to check whether a state-machine representation of a system is a model of, or satisfies, a formula in temporal logic. Temporal logic allows for the specification of critical system properties…" (col. 2, lines 30-37). This is the traditional definition of model checking, and Cleaveland

et al. speak of the "state machine" in the conventional way: as the model of the design itself, which is tested against the rules of the specification. Cleaveland et al. make no mention or suggestion of creating a tableau from the specification, whether in the sense of a state machine, as recited in claim 1, or a maximal proof tree as defined by Cleaveland.

Returning now to claim 1, the claim recites two separate entities: a model and a tableau, each with its own set of states and transitions. Although Cleaveland and Cleaveland et al. may relate to model states and model transitions, neither of these references teaches or suggests the creation of a tableau with tableau states and tableau transitions. Thus, the references certainly cannot be taken to suggest the comparison of tableau transitions with model transitions, as required by claim 1. Therefore, claim 1 is believed to be patentable over the cited art. In view of the patentability of claim 1, claims 2-18, which depend from claim 1, are also believed to be patentable.

Independent claims 19 and 21 respectively recite a verification processor and a computer software product, which operate on principles similar to the method of claim 1. Therefore, for the reasons explained above with reference to claim 1, claims 19 and 21 are believed to be patentable, as are claim 20, which depends from claim 19, and claims 22-24, which depend from claim 21.

Claim 25 recites a method for verification that comprises providing an implementation model of a target system, and creating a tableau from a specification of the system, as in the method of claim 1. The tableau defines tableau states with tableau transitions between the tableau states in accordance with the properties in the specification. The model and the tableau are compared by inputting the model and the tableau to an automatic model checking program.

As explained above with reference to claim 1, Cleaveland uses the term "tableau" to refer to a hierarchy of rules. Neither Cleaveland nor Cleaveland et al. makes any mention or suggestion of tableau states and tableau transitions, in addition to model states and model transitions, as required by claim 25. Therefore, claim 25 is believed to be patentable over the cited art, as are claims 26-30, which depend from claim 25.

Claims 31 and 32 respectively recite model checking apparatus and a computer software product, which operate on principles similar to the method of claim 25. Therefore, for the reasons explained above with reference to claim 25, claims 31 and 32 are also believed to be patentable.

Applicant believes the remarks stated above to be fully responsive to all the grounds of rejection cited by the Examiner. In view of these remarks, all the claims in the present patent application are believed to be in condition for allowance. Prompt notice to this effect is requested.

Respectfully submitted,

S. Peter Ludwig
Registration No. 25,351
Attorney for Applicant

Darby & Darby P.C.
P.O. Box 5257
New York, NY 10150-5257
(212) 527-7700